Icahn
School of
Medicine at
Mount
Sinai

Introduction to coding with R

Part I

Joselyn Chavez

04/12/2022

Data structures in R

- Vectors
- Matrices
- Data frames
- Lists
- Functions

Vectors

Creating a vector

Using the assignment operator

For one value

```
my_vector <- 10
my_vector <- "a"
```

Using the combine function

For two or more values

```
my_vector <- c(1,10,25,30)
my_vector
```

```
## [1]  1 10 25 30
```

```
my_vector <- c("a","b","c")
my_vector
```

```
## [1] "a" "b" "c"
```

Using the seq() function

```
my_vector <- seq(1:10)
my_vector
```

```
##  [1]  1  2  3  4  5  6  7  8  9 10
```

```
my_vector <- seq(from = 0, to = 10, by = 2)
my_vector
```

```
## [1]  0  2  4  6  8 10
```

Vector features

- Vectors have only one dimension (length)

```
my_vector <- c(1,2,3,4)
length(my_vector)
```

```
## [1] 4
```

- All vector components must be the same type
  - Numeric
  - Integer
  - Double
  - Character
  - Factor
  - Logical

- Numeric

```
x_num <- c(1, 2, 3)
class(x_num)
```

```
## [1] "numeric"
```

- Integer

```
x_int <- c(1L, 2L, 3L)
class(x_int)
```

```
## [1] "integer"
```

- Double

```
x_dbl <- c(1, 2.5, 3.1)
typeof(x_dbl)
```

```
## [1] "double"
```

- Character

```
x_chr <- c("a", "b", "c")
class(x_chr)
```

```
## [1] "character"
```

- Factor

```
x_fct <- factor("a","b","c")
class(x_fct)
```

```
## [1] "factor"
```

- Logical

```
x_log <- c(TRUE, FALSE, TRUE)
class(x_log)
```

```
## [1] "logical"
```

- R finds a way to unify data type when there is more than one per vector

```
x <- c(1, "a", TRUE)
x
```

```
## [1] "1"    "a"    "TRUE"
```

```
class(x)
```

```
## [1] "character"
```

Missing values

- NA

```
x <- c(1, "a", TRUE, NA)
x
```

```
## [1] "1"    "a"    "TRUE" NA
```

- NaN

```
x <- c(10, -1, NA)
log(x)
```

```
## [1] 2.302585      NaN       NA
```

How to access vector elements?

Using integer as index

Vector index in R starts from 1

```
x <- c(1,2,3,4,5)
x
```

```
## [1] 1 2 3 4 5
```

```
x[3] # Extract the third element
```

```
## [1] 3
```

```r
x <- c(1,2,3,4,5)

# Extract index from 3 to 5
x[3:5]
```

```
## [1] 3 4 5
```

```r
x <- c("a","b","c","d","e")

# Extract index 2 and 5
x[c(2,5)]
```

```
## [1] "b" "e"
```

Using the name as index

```
x <- c(1,3,10)
names(x)
```

```
## NULL
```

```
x <- c("first"= 1,"second"=3,"third"=10)
x
```

```
##  first second  third
##      1      3     10
```

```
names(x)
```

```
## [1] "first"  "second" "third"
```

```r
x <- c(1,3,10)

names(x) <- c("first", "second", "third")
x
```

```
##  first second  third
##      1      3     10
```

```r
x["second"]
```

```
## second
##      3
```

```r
x[c("first","third")]
```

```
## first third
##     1    10
```

Using logical evaluation as index

```
x <- seq(1:10)
x
```

```
##  [1]  1  2  3  4  5  6  7  8  9 10
```

```
x < 5
```

```
##  [1]  TRUE  TRUE  TRUE  TRUE FALSE FALSE FALSE FALS
```

```
x[x < 5]
```

```
## [1] 1 2 3 4
```

```r
x <- c("a","a","b","c","c","c")

x == "c"
```

```
## [1] FALSE FALSE FALSE  TRUE  TRUE  TRUE
```

```r
x[x == "c"]
```

```
## [1] "c" "c" "c"
```

Excercise

Using the vector:

```
fruits <- c("apples" = 1,
            "cherries" = 10,
            "mangos" = 3)
```

Extract the number of cherries using 1) the integer index, 2) the name, and 3) a logical evaluation

How to modify a vector?

- Adding a new element

```
x <- c("a","b","c")
x
```

```
## [1] "a" "b" "c"
```

```
x[4] <- "d"
x
```

```
## [1] "a" "b" "c" "d"
```

- Removing an element

```
x
```

```
## [1] "a" "b" "c" "d"
```

```
x[-2]
```

```
## [1] "a" "c" "d"
```

```
x <- x[-2]
x
```

```
## [1] "a" "c" "d"
```

- Replacing an element

```
x
```

```
## [1] "a" "c" "d"
```

```
x[x == "d"] <- "e"
x
```

```
## [1] "a" "c" "e"
```

```
x[1] <- "m"
x
```

```
## [1] "m" "c" "e"
```

- Selecting and replacing elements using negative conditional

```
x <- c("a","a","b","c","c","c")
x[x != "c"]
```

```
## [1] "a" "a" "b"
```

```
x[x != "c"] <- "e"
x
```

```
## [1] "e" "e" "e" "c" "c" "c"
```

Excercise

Using the vector:

```
fruits <- c("apples" = 1,
            "cherries" = 10,
            "mangos" = 7)
```

- Add a new fruit to the vector
- Select all fruits with values bigger than 5
- Replace the apples number

Operations with vectors

Arithmetic operations

```
x <- c(1,2,3,4,5)
x + 1
```

```
## [1] 2 3 4 5 6
```

```
x*3
```

```
## [1]  3  6  9 12 15
```

```
log10(x)
```

```
## [1] 0.0000000 0.3010300 0.4771213 0.6020600 0.69897
```

Operations between vectors

```
x <- c(1,2,3,4,5)
y <- c(6,7,8,9,10)
x + y
```

```
## [1]  7  9 11 13 15
```

```
x <- c(1,2,3,4,5)
y <- c(1,2)
x + y
```

```
## Warning in x + y: longer object length is not a mul
## length
```

```
## [1] 2 4 4 6 6
```

Matrices

Creating a matrix

Matrices are objects with elements arranged in a two-dimensional layout.

```
matrix(data = 1:12, nrow = 4, ncol = 3)
```

```
##      [,1] [,2] [,3]
## [1,]    1    5    9
## [2,]    2    6   10
## [3,]    3    7   11
## [4,]    4    8   12
```

By deafult, elements are arranged by column

Arranging the matrix by row

```
matrix(data = 1:12,
       nrow = 4, ncol = 3,
       byrow = TRUE)
```

```
##      [,1] [,2] [,3]
## [1,]    1    2    3
## [2,]    4    5    6
## [3,]    7    8    9
## [4,]   10   11   12
```

Properties of matrices

- nrow

```
my_matrix <- matrix(data = 1:12,
                    nrow = 4, ncol = 3)
nrow(my_matrix)
```

```
## [1] 4
```

- ncol

```
ncol(my_matrix)
```

```
## [1] 3
```

- dim

```
dim(my_matrix)
```

```
## [1] 4 3
```

- rownames

```
rownames(my_matrix)
```

```
## NULL
```

- colnames

```
colnames(my_matrix)
```

```
## NULL
```

Assigning rownames and colnames

```
rownames(my_matrix) <- c("A","B","C","D")
colnames(my_matrix) <- c("a","b","c")

my_matrix
```

```
##   a b  c
## A 1 5  9
## B 2 6 10
## C 3 7 11
## D 4 8 12
```

How to access matrix elements?

Using row and column index

Syntaxis: matrix[row,column]

```
my_matrix
```

```
##   a b  c
## A 1 5  9
## B 2 6 10
## C 3 7 11
## D 4 8 12
```

```
my_matrix[2,3]
```

```
## [1] 10
```

Select rows 1 to 2 from column 3

```
my_matrix[1:2,3]
```

```
##  A  B
##  9 10
```

Select rows 1 to 2 from columns 2 to 3

```
my_matrix[1:2,2:3]
```

```
##   b  c
## A 5  9
## B 6 10
```

Select all rows from column 2

```
my_matrix[,2]
```

```
## A B C D
## 5 6 7 8
```

Select all columns from row 2

```
my_matrix[2,]
```

```
##  a  b  c
##  2  6 10
```

Operations with matrices

Arithmetic operations

```
my_matrix + 10
```

```
##    a  b  c
## A 11 15 19
## B 12 16 20
## C 13 17 21
## D 14 18 22
```

```
my_matrix * 2
```

```
##   a  b  c
## A 2 10 18
## B 4 12 20
## C 6 14 22
## D 8 16 24
```

Operations with matrices

```
matrix1 <- matrix(1:6, nrow = 2, ncol = 3)
matrix2 <- matrix(7:12, nrow = 2, ncol = 3)
```

```
matrix1
```

```
##      [,1] [,2] [,3]
## [1,]    1    3    5
## [2,]    2    4    6
```

```
matrix1 + matrix2
```

```
##      [,1] [,2] [,3]
## [1,]    8   12   16
## [2,]   10   14   18
```

```
matrix2
```

```
##      [,1] [,2] [,3]
## [1,]    7    9   11
## [2,]    8   10   12
```

Thanks!

Ilustration by Allison Horst